TR A4687.01

PONG.
The ELCOM net-watch procedure for TCP/IP networks.

Birger Stene
Convener

May 2008

SINTEF

SINTEF Energiforskning AS

Postadresse: 7465 Trondheim
Resepsjon: Sem Sælands vei 11
Telefon: 73 59 72 00
Telefaks: 73 59 72 50

www.energy.sintef.no

Foretaksregisteret:
NO 939 350 675 MVA

TECHNICAL REPORT

SUBJECT/TASK (title)

PONG

The ELCOM net-watch procedure for TCP/IP networks

CONTRIBUTOR(S)
ELCOM Working Group
Convener Birger Stene

CLIENTS(S)
Joint project: ABB, Siemens AS, SINTEF Energy Research, Statnett SF

TR NO.	DATE	CLIENT'S REF.	PROJECT NO.
TR A4687.01	2008-05-01	A Sveen, N-J Aulie, B Stene, A Larsen	12X513

ELECTRONIC FILE CODE	RESPONSIBLE (NAME, SIGN.)	CLASSIFICATION
	Birger Stene	Open

ISBN N0.	REPORT TYPE	RESEARCH DIRECTOR (NAME, SIGN)	COPIES	PAGES
82-594-1236-5		Petter Støa	12	10

DIVISION	LOCATION	LOCAL FAX
Energy Systems	Sem Sælands vei 11, 7465 Trondheim	+47 73 59 72 50

RESULT (summary)

This document is one of a series of technical reports which form the complete ELCOM-90 documentation. This is version .01 of the report with minor changes regarding responsible people and references. Future updates and new versions will NOT be published for this reason. New versions will only be submitted when technical changes are made.
Please see SINTEF's homepage at: http://www.sintef.no/ELCOM-90. From here you can download the latest version of all relevant documents as pdf-files for free.

This report provides a description of a mechanism for monitoring ELCOM-90 connections on a TCP/IP network, using a keep alive request response dialog when the connection is otherwise alive. This function, PONG, is integrated in the present ELCOM-90 reference version. It is recommended to use PONG instead of a mechanism based on ICMP echo datagrams as used in the UNIX command PING.

The use of ELCOM-90 Test Association Functional Unit is still recommended even if PONG is used to monitor the network, as it enable supervision of the remote user element as well.

KEYWORDS

SELECTED BY AUTHOR(S)		
	Communication protocol	Network monitoring
	Control centres	ELCOM-90

TABLE OF CONTENTS

Page

1 INTRODUCTION

This document describes a mechanism for monitoring ELCOM-90 connections on TCP/IP, using a keep-alive request/response dialog when the connection is otherwise idle.

The reference version of ELCOM-90 has had a mechanism for supervising remote TCP/IP partners based on Internet Control Message Protocol (ICMP) echo datagrams, as used in the unix command ping. Hence the name 'PONG' for this new approach.

2 ASSOCIATED DOCUMENTS

2.1 ELCOM-90 documentation

The ELCOM-90 documentation set consists of the following individual documents, referred to by this document:
References from [1] to [7] concerns ELCOM-83.

This document is one of a series of technical reports which form the complete ELCOM-90 documentation. Below you will find the numbers and titles for all the associated technical reports. New versions may be submitted when technical changes are made.
Please see SINTEF's homepage at: http://www.sintef.no/ELCOM-90. From here you can download the latest version of all relevant documents as pdf-files for free.

[8]: TR 3701: **ELCOM-90 Application Programming Interface Specification**

[9]: TR 3702: **ELCOM-90 Application Service Element. Service Definition**

[10]: TR 3703: **ELCOM-90 Application Service Element. Protocol Specification**

[11]: TR 3704: **ELCOM-90 Presentation Programming Interface Specification**

[12]: TR 3705: **ELCOM-90 Presentation Service Definition**

[13]: TR 3706: **ELCOM-90 Presentation Protocol Specification**

[14]: TR 3825: **ELCOM-90 User Element Conventions**

[15]: TR A3933: **ELCOM-90 Local Conventions**

[16] TR A4687: **PONG. The ELCOM net-watch procedure for TCP/IP networks**

[17] TR A4124: **ELCOM-90 Application Service Element, User's manual.**

[18] TR A6196: **Securing ELCOM-90 with TLS.**

3 TECHNICAL BACKGROUND AND HISTORY

When using the ELCOM-90 protocol for real-time data transfer, such as commands and unsolicited data transfer, it is important to be able to detect network errors in a timely fashion, typically to fail over the connection to use a different route.

When TCP/IP is used as the transport, as is more and more common, the network layers themselves does not offer adequate functionality for this, and some errors in the network may go undetected for minutes or even longer, in particular on ELCOM connections that may be idle, such as an unsolicited data transfer channel.

The following sections describe different methods for supervising ELCOM on TCP/IP, some of which are still relevant, and some that have been used in the past, ending with the solution described in detail in this document.

3.1 ELCOM TEST ASSOCIATION

The ELCOM Test Association FU, commonly known as test connect, described in section 5.8 in [14], allows the Initiator User Element to monitor its connections with remote partners (provided the remote User Element implements the FU).
This function provides a good test for the availability of a particular ELCOM connection, as it tests all the way up to the User Element at the remote end.

The Test Connect does not, however, provide any means for the Responder to actively monitor the status of a connection. It is of course possible to apply local conventions between partners, requiring the initiator to perform test connects at predefined intervals, and using this to indirectly monitor the initiator from the responder, but this may be difficult to coordinate.

Another issue with the test connect is that the timeout value is generally 2 minutes, and this is not easily changed. Some ELCOM users desire shorter timeouts to enable faster detection of network errors.

As a conclusion, the use of ELCOM Test Connect is still recommended even if PONG is used to monitor the network, as it enables supervision of the remote user element as well.

3.2 TCP KEEPALIVE

The TCP keep alive mechanism is available in most TCP/IP implementations to enable termination of failed connections even in the absence of traffic. Whereas the general ideas are much the same as what is described below for PONG, there are problems with using this:

- most TCP/IP implementations have a system wide timer value for this, typically with a default on the order of about 2 hours.
- changing this may affect other TCP/IP programs in unforeseen ways.
- tests using this mechanism at EFI in the early 90s failed to give satisfactory results on all platforms.

3.3 ICMP ECHO (PING)

A monitoring mechanism based on ICMP echo datagrams, as used by the ‘ping’ command has been available in the reference version of the ELCOM-90 provider for a while. The basic design is to identify the remote systems being connected, and using ICMP echo to test whether these systems are alive. Deployment of this solutions has not been entirely successful, and current trends in networking is rendering this solution undesirable; in particular the fact that ELCOM-90 now often passes a one or more firewalls in a typical network, and many of these are configured to block ICMP datagrams.

Although this function has the virtue of requiring no supporting code in the remote end, it is not recommended, and is now considered unsupported in the reference version.

3.4 CONNECTION-LEVEL KEEPALIVE (PONG)

Based on the experience from the ping function the PONG functions were designed with the following features:

- Monitors each connection individually. This makes implementation easier at the cost of some bandwidth.

- Uses any traffic to monitor, causing keep-alive messages to be sent only when a connection is idle for some time.

- Uses redundant fields in the TCP/IP address format to perform a handshake at connection, so that a PONG-enabled provider will work with existing providers without requiring configuration of PONG on a per partner level.

The following chapter describes the implementation of this function.

4 IMPLEMENTATION DETAILS

4.1 OVERVIEW AND MINIMUM REQUIREMENTS

The PONG function is such that it allows some freedom in how it is implemented in a provider implementation. Since it is dependent on cooperation with the remote part to achieve its end, some minimum functionality is, however, required:

- The handshake mechanism described in the next section must be implemented

- When a keep-alive request is received, a keep-alive response must be sent to the originating partner immediately

To be useful for its own user elements, a provider will of course also need to implement a mechanism for sending the keep-alive requests to a partner. Some configurable timeout mechanism is needed, to determine when to do this, as well as a mechanism to signal the failure of a connection to the user element in question, typically by forcing an abort.

It is also recommended that the transmission of keep-alive requests are minimized, by only sending such requests when the connection in question is idle for some time.

4.2 CONNECTION HANDSHAKE

The connection handshake is used to identify whether a remote partner supports PONG or not, as a connection is established. Consider the ELCOM-90 TCP/IP address format shown below. This structure is as it is, as it is based on the memory layout of a sockaddr_in structure. In particular, the AF_INET field will have varying contents based on the byte order of local machine hardware, since this is just a constant for the local socket implementation. Thus it is expected that the remote part must ignore the contents of received AF_INET, and this field was selected to perform a handshake for the PONG function.

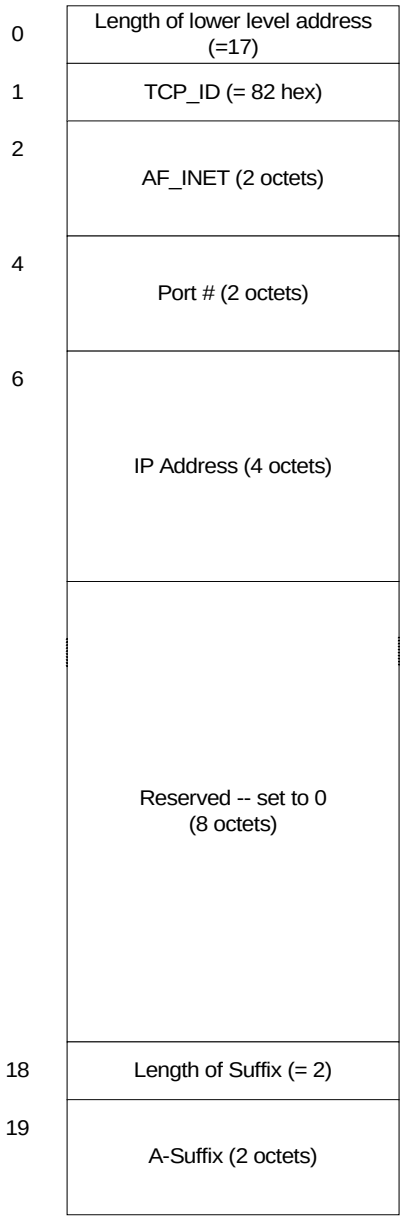

Figure 1 ELCOM-90 TCP/IP address format.

The AF_INET field consists of two bytes (octets). For packets originating from partners without PONG, these bytes will be 0,2 or 2,0, depending on the byte order (the value of AF_INET). For the handshake, second byte of AF_INET is used, such that:

- In the connect request, an initiator with PONG sets the second byte of AF_INET in the initiator address to 100.

- In the connect response, a responder with PONG sets the second byte of AF_INET in the responder address to 100.

- It is expected that a provider that does not support PONG will ignore the AF_INET in the initiator address (as it should), and deliver a value of 0 or 2 in the second byte of the responder AF_INET, as it will if it actually places the constant AF_INET in this field.

- The Initiator will not use PONG for this partner unless a positive handshake is received.

4.3 KEEPALIVE REQUEST

The keep-alive request contains a single octet with the value 8 (00001000 in binary), in addition to the two-byte length always present when using Elcom on TCP/IP (the length field contains the value 1).

4.4 KEEPALIVE RESPONSE

The keep-alive response contains a single octet with the value 12 (00001100 in binary), in addition to the two-byte length always present when using Elcom on TCP/IP (the length field contains the value 1).

4.5 TIMING CONSIDERATIONS

The timing aspects of this function can be described using two timeouts:

- The idle timeout (T1) – the time period that a connection must be idle before a keep-alive request is sent.
- The response timeout (T2) – the time period allowed before a keep-alive response (or any other data) must have been received after a keep-alive request is sent.

The error detection time will be > T2 and <= T1 + T2, assuming T1 is reset whenever data is received. Note that the in the reference implementation a single timer is used, i.e. T1 = T2; see below for a description of this.

Note that regardless of timeout semantics, due consideration needs to be made towards the load both on the network and on the local machine. Setting the timeouts too low will invariably lead to service interruption from false timeouts, caused not by network failure, but by high load, on the network or on the local system.

4.6 IMPLEMENTATION IN THE REFERENCE VERSION

The reference version of the ELCOM-90 (the current version) contains an implementation of the PONG function with the following characteristics.

4.6.1 Timeout handling

The timeout handling uses a single timeout value, set by the key PONG_TIMER in elc-conf. This can be set to 0 to disable the PONG function, and a minimum value of 5 seconds is currently enforced.

This timeout value controls a check function, which will inspect all active connections to PONG-enabled partners each time the timer fires. If a connection has been idle for two consecutive checks, a keep-alive request is sent. If no response to this is received (or other traffic) before the next check, the connection is failed.

This gives a detection time which is >2 * PONG_TIMER and <3 * PONG_TIMER.

4.6.2 Reporting and logging

Once a connection is failed, an abort is generated to the local user element, with a result code of 41, “Disconnected by the network layer”. In addition to this, a message is logged to the standard error output of the provider (normally redirected to a log file).